WHAT IS CLAIMED IS:

1. A system, comprising:

data capture logic configured to capture data events from a nondeterministic data bus;

a system memory including a plurality of addressable locations, wherein a subset of said plurality of addressable locations is configured as a data event buffer;

a direct memory access (DMA) transfer engine coupled to said data capture logic and to said system memory and configured to perform a DMA transfer operation of said captured data events from said data capture logic to a region of said data event buffer as portions of said captured data events become available from said data capture logic; and

an application configured to access said data event buffer to process said captured data events without said DMA transfer operation being stopped;

wherein in response to said region of said data event buffer being filled, said DMA transfer engine is further configured to perform said DMA transfer operation to a different region of said data event buffer without said DMA transfer operation being stopped.

2. The system as recited in claim 1, wherein said data event buffer is configured as a circular data event buffer.

3. The system as recited in claim 1, wherein said data event buffer is configured as a linear data event buffer.

4. The system as recited in claim 1, wherein said data event buffer is allocated within a kernel address space.

5. The system as recited in claim 1, wherein said nondeterministic data bus conforms to the IEEE-488 General Purpose Interface Bus (GPIB) standard.

6. The system as recited in claim 1, wherein said data capture logic is further configured to assign a respective sample index value to each of said captured data events, and wherein said DMA transfer engine is further configured to pause said DMA transfer operation in response to detecting that the respective sample index value corresponding to one of said captured data events matches a selected sample index value.

7. The system as recited in claim 1, wherein said data capture logic is further configured to store said captured data events in a capture buffer, wherein in response to detecting an overflow of said capture buffer, said data capture logic is further configured to stop capturing data events without said DMA transfer operation being stopped.

8. The system as recited in claim 1, wherein said application is further configured to display said processed data events substantially in real time with respect to the occurrence of the corresponding captured data events on said nondeterministic data bus.

9. The system as recited in claim 1, wherein said DMA transfer engine is further configured to convey an indication of data readiness to said application after transferring a given number of said captured data events to said data event buffer.

10. The system as recited in claim 1, wherein said application is further configured to request an indication of data readiness after a given period of time has elapsed without receiving a signal indicative of data readiness.

11. A method, comprising:

capturing data events from a nondeterministic data bus;

transferring said captured data events to a region of a data event buffer as portions of said captured data events become available;

accessing said data event buffer to process said captured data events without stopping said transferring;

detecting that said region of said data event buffer is full during said transferring; and

in response to said detecting that said region is full, transferring said captured data events to a different region of said data event buffer without stopping.

12. The method as recited in claim 11, wherein said transferring comprises a DMA transfer operation.

13. The method as recited in claim 11, wherein said data event buffer is configured as a circular data event buffer.

14. The method as recited in claim 11, wherein said data event buffer is configured as a linear data event buffer.

15. The method as recited in claim 11, wherein said data event buffer is allocated within a kernel address space.

16. The method as recited in claim 11, wherein said nondeterministic data bus conforms to the IEEE-488 General Purpose Interface Bus (GPIB) standard.

17. The method as recited in claim 11, further comprising:

assigning a respective sample index value to each of said captured data events prior to transferring each said captured data event;

prior to transferring a given captured data event, detecting that said respective sample index value corresponding to said given captured data event matches a selected sample index value; and

in response to detecting said match, pausing said transferring of said captured data events.

18. The method as recited in claim 11, further comprising:

storing said captured data events in a capture buffer prior to said transferring;

detecting an overflow of said capture buffer; and

in response to detecting said overflow, stopping said capturing of said data events without stopping said transferring of said captured data events.

19. The method as recited in claim 11, further comprising displaying said processed data events substantially in real time with respect to the occurrence of the corresponding captured data events on said nondeterministic data bus.

20. The method as recited in claim 11, further comprising conveying an indication of data readiness to an application after transferring a given number of said captured data events to said data event buffer.

21. The method as recited in claim 11, further comprising requesting an indication of data readiness of said captured data events in said data event buffer after a given period of time has elapsed without receiving a signal indicative of data readiness.

22. A computer-accessible medium comprising program instructions, wherein said program instructions are computer-executable to:

enable data capture logic configured to capture data events from a nondeterministic data bus;

configure a direct memory access (DMA) transfer engine to perform a DMA transfer operation of said captured data events to a region of a data event buffer as portions of said captured data events become available; and

access said data event buffer to process said captured data events without said DMA transfer operation stopping;

wherein in response to said region of said data event buffer being filled, said DMA transfer engine is further configured to perform said DMA transfer operation to a different region of said data event buffer without said DMA transfer operation being stopped.

23. The computer-accessible medium as recited in claim 22, wherein said data event buffer is configured as a circular data event buffer.

24. The computer-accessible medium as recited in claim 22, wherein said data event buffer is configured as a linear data event buffer.

25. The computer-accessible medium as recited in claim 22, wherein said data event buffer is allocated within a kernel address space.

26. The computer-accessible medium as recited in claim 22, wherein said nondeterministic data bus conforms to the IEEE-488 General Purpose Interface Bus (GPIB) standard.

27. The computer-accessible medium as recited in claim 22, wherein said data capture logic is further configured to assign a respective sample index value to each of said captured data events, and wherein said DMA transfer engine is further configured to pause said DMA transfer operation in response to detecting that the respective sample index value corresponding to one of said captured data events matches a selected sample index value.

28. The computer-accessible medium as recited in claim 22, wherein said data capture logic is further configured to store said captured data events in a capture buffer, wherein in response to detecting an overflow of said capture buffer, said data capture logic is further configured to stop capturing data events without said DMA transfer operation being stopped.

29. The computer-accessible medium as recited in claim 22, wherein said program instructions are further executable to display said processed data events substantially in real time with respect to the occurrence of the corresponding captured data events on said nondeterministic data bus.

30. The computer-accessible medium as recited in claim 22, wherein said DMA transfer engine is further configured to convey an indication of data readiness to an

application after transferring a given number of said captured data events to said data event buffer.

31. The computer-accessible medium as recited in claim 22, wherein said program instructions are further executable to request an indication of data readiness after a given period of time has elapsed without receiving a signal indicative of data readiness.